

07-03-00

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventorship..... Kabir et al.
 Applicant..... Microsoft Corporation
 Attorney's Docket No. MS1-569US
 Title: Systems and Methods for Switching Internet Contexts Without Process Shutdown

TRANSMITTAL LETTER AND CERTIFICATE OF MAILING

To: Commissioner of Patents and Trademarks,
 Washington, D.C. 20231

From: James R. Banowsky (Tel. 509-324-9256; Fax 509-323-8979)
 Lee & Hayes, PLLC
 421 W. Riverside Avenue, Suite 500
 Spokane, WA 99201

The following enumerated items accompany this transmittal letter and are being submitted for the matter identified in the above caption.

1. Specification—title page, plus 31 pages, including 33 claims and Abstract
2. Transmittal letter including Certificate of Express Mailing
3. 6 Sheets Formal Drawings (Figs. 1-6)
4. Return Post Card

Large Entity Status ☒ [x]

Small Entity Status ☐ []

Date: 6/30/00

By: James R. Banowsky

James R. Banowsky
 Reg. No. 37,773

CERTIFICATE OF MAILING

I hereby certify that the items listed above as enclosed are being deposited with the U.S. Postal Service as either first class mail, or Express Mail if the blank for Express Mail No. is completed below, in an envelope addressed to The Commissioner of Patents and Trademarks, Washington, D.C. 20231, on the below-indicated date. Any Express Mail No. has also been marked on the listed items.

Express Mail No. (if applicable) EL624352913

Date: 6/30/2000

By: Dana L. Calhoun

Dana L. Calhoun

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Systems and Methods for Switching Internet Contexts
Without Process Shutdown**

Inventor(s):

Ahsan S. Kabir

Erik Snapper

Darren Mitchell

Rajeev Dujari

ATTORNEY'S DOCKET NO. MS1-569US

TECHNICAL FIELD

This invention relates to managing network connections and, more particularly, to switching Internet contexts without requiring process shutdown.

BACKGROUND

When a user browses the Internet using a browser on a computer, the user builds up an Internet context for her browsing activity, e.g., through cookies, history and personalized data. As the popularity of Internet browsing has grown, it is not uncommon to have more than one person using a single computer. When different people use the same computer for Internet browsing, Internet contexts can become confused. As a result, one user may no longer know which websites he has visited, and users' preferences for the same website may conflict.

To accommodate multi-user arrangements, operating systems support "profiles" for different users, each profile defining a particular work environment configured for a particular user. Each user of a computer can maintain his or her own profile, which accommodates the particular user. However, switching from one Internet context to another requires a system-wide process shutdown, meaning that a current user must end any tasks that are in progress and log off before a new user can log in. In addition, most families do not go to the trouble to enable profiles on their home computers and, therefore, they cannot switch contexts at all.

SUMMARY

The implementations described herein allow web browsers and other Internet client applications to provide separate contexts for different users, or identities without requiring a process shutdown. This is accomplished by creating an Internet context management component that can select new directory locations for storing non-content state and by tagging content to indicate whether or not the data is personalized to a specific user.

In one implementation, an Internet management object maintains a set of objects called containers, one for each class of Internet state. Each container is located in its own directory on the file system of a computer. This set of containers is called a cache. The information in these containers forms an Internet context that is associated with an identity, each computer user having a unique identity.

When a client wants to change Internet contexts, it supplies a globally unique identifier (guid) that denotes a specific identity to the Internet management object. This initiates the process by which the Internet management object will switch Internet contexts.

The Internet management object shuts down the cache for a current user to prevent any operations from utilizing the cache any further. To simplify naming directories and content tagging, the Internet management object associates each guid with an ordinal according to one described implementation. The Internet management object attempts to create and use a set of containers associated with the new guid using this ordinal. If this is unsuccessful, the Internet management object will restart the cache and continue to use the current Internet context.

1 Otherwise, the Internet management object flushes any authentication credentials,
2 purges session cookies and resets the session start time. In other words, the
3 Internet browser or client application will behave as if it has been closed and
4 restarted.

5 As the Internet context is built with the new identity, cookies, history and
6 other non-content for the new context are placed in a directory associated with the
7 ordinal that maps to the identity's guid. However, content cache is handled
8 differently.

9 To avoid duplicating Internet content across multiple directories (and thus
10 using large amounts of disk space) content is shared across all identities. But there
11 are cases in which an item is meant for only one identity and there are cases
12 wherein the same universal resource locator (URL) will map to different content
13 for different identities. For example, if a web page is a secure page that only some
14 users can access, or if a page uses cookies, etc.

15 When content is shared, it is stored in a common directory that is used by
16 each identity. This directory contains an index file that has multiple records
17 indicating the location and other information for one or more web pages. When
18 information from a web page is stored, the URL for the web page is hashed and
19 the hash value is used to look up an index record. The index record reveals
20 information regarding this web page data, including a file name where the
21 information is stored.

22 When content is user-specific, information indicating that the content is not
23 to be shared is provided to the Internet management object. When the Internet
24 management object detects that this indication has been provided, it performs the
25 look up by hashing a combination of the URL and the ordinal associated with the

1 appropriate identity. This locates a unique user-specific index record similar to
2 that used for shared-content, except that the information is specific for the user,
3 which only the proper identity can access the private information stored therein.
4

5 **BRIEF DESCRIPTION OF THE DRAWINGS**

6 A more complete understanding of exemplary methods and arrangements
7 of the present invention may be had by reference to the following detailed
8 description when taken in conjunction with the accompanying drawings wherein:

9 Fig. 1 is a diagram of an exemplary computer system on which the
10 described embodiments may be implemented.

11 Fig. 2 is a flow diagram of a method according to a general implementation
12 in which an Internet context is changed from an Internet context for a first identity
13 to an Internet context for a second identity.

14 Fig. 3 is a block diagram of a computer having memory storage.

15 Fig. 4 is a depiction of an index file having multiple records.

16 Fig. 5 is a simplified diagram of a registry in the system of Fig. 3.

17 Fig. 6 is a flow diagram illustrating a method for handling shared and user-
18 specific Internet content.
19
20
21
22
23
24
25

DETAILED DESCRIPTION

The invention is illustrated in the drawings as being implemented in a suitable computing environment. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, to be executed by a computing device, such as a personal computer or a hand-held computer or electronic device. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Exemplary Computer Environment

The various components and functionality described herein are implemented with a number of individual computers. Fig. 1 shows components of typical example of such a computer, referred by to reference numeral 100. The components shown in Fig. 1 are only examples, and are not intended to suggest any limitation as to the scope of the functionality of the invention; the invention is not necessarily dependent on the features shown in Fig. 1.

information such as computer-readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 100, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Fig. 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

information into the computer 100 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball, or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port, or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 195.

The computer may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 100, although only a memory storage device 181 has been illustrated in Fig. 1. The logical connections depicted in Fig. 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When used in a LAN networking environment, the computer 100 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 100 typically includes a modem

172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 100, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Fig. 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Fig. 2 is a flow diagram of a method according to a general implementation in which an Internet context is changed from an Internet context for a first identity to an Internet context for a second identity. At step 200, a client computer operates within a context for a first identity. An Internet context comprises data associated with a particular user, or identity. Such data includes a history of web sites that a particular user has visited, cookies that have been placed on the computer by a particular user that allows a web site to gather information about that user, web page content, etc.

At step 202, a request is received to switch from the first identity to a second identity. The request is received by an Internet management object operating on the client computer. The Internet management object is responsible for maintaining an Internet connection and managing the data associated therewith. Although described herein as an Internet management object, an object may be more generally described as a network management object, which performs similar task on networks other than the Internet, such as an intranet, a local area network, a wide area network, etc.

The registry 308 is a global database that includes multiple profiles of configuration data as well as a pointer that points to a profile currently in use. The registry 308 includes identity 0 312, identity 1 314, identity 2 316 and a current

Wininet 310 contains pointers to containers that make up an Internet context. Wininet 310 includes a cookies pointer 320, a history pointer 322, a TIF (temporary Internet files) pointer 324, and a user-defined pointer 326. It is noted that there may be more than one user-defined container and pointer thereto. However, for discussion purposes, only one user-defined pointer is shown herein.

Wininet 310 also includes parameters associated with Internet browsing, such as a session start time 328 and authentication credentials 330. It is noted that these parameters are exemplary only and there may be other or additional parameters, depending on the implementation.

The Wininet pointers reference memory locations in the hierarchical directory structure 306. The cookies pointer 320 references a cookies directory 332; the history pointer 322 references a history directory 334; the TIF pointer 324 references a TIF directory 336; and the user-defined pointer 326 references a user-defined directory 338. It is noted that the cookies directory 332, the history directory 334, the TIF directory 336 and the user-defined directory 338 are subdirectories of at least one superior directory in the directory hierarchy. However, superior directories in the structure are not shown for convenience purposes.

1 HKEY_USERS subkeys (profiles) that is currently in use.
2 HKEY_CURRENT_USER 508 contains numerous subkeys 510, one of which is
3 the IDENTITIES subkey 512. The IDENTITIES subkey 512 contains a
4 CURRENT_IDENTITY 514 and an individual subkey 516 and 518 for each
5 configured identity. In this example, IDENTITY 1 is identity 1 314 in Fig. 3 and
6 IDENTITY 2 is identity 2 316 in Fig. 3. CURRENT_IDENTITY 514 is a value
7 that points to the one of these two identities 516, 518 that is currently in use. If no
8 identity is selected for use, then the system is in a "non-identity state" and
9 CURRENT-IDENTITY subkey 514 points to no identity.

10 There is also a TIF subkey 520, a COOKIES subkey 522, a HISTORY
11 subkey 524 and a USER-DEFINED subkey 526. The TIF subkey 520 contains a
12 value that points to a location of Internet content, or temporary internet files, for
13 the CURRENT_IDENTITY 514. The COOKIES subkey 522 contains a value that
14 points to a location of cookies for the CURRENT_IDENTITY 514. The
15 HISTORY subkey 524 contains a value that points to a location of history data for
16 the CURRENT_IDENTITY 514. The USER-DEFINED subkey 526 contains a
17 value that points to a location of user-defined data for the CURRENT_IDENTITY
18 514. It is noted that there may be more than one USER-DEFINED subkey, or
19 there may be no USER-DEFINED subkey depending on the implementation. For
20 the present discussion, only one USER-DEFINED subkey is shown.

21 When a switch request is submitted, it is determined if the identity is a new
22 identity or if it has been used before. If the identity has not been used before, then
23 the registry subkey values for COOKIES 522, HISTORY 524 and USER-
24 DEFINED 526 are set to new values and an Internet context begins to be built for
25 the new user. An ordinal x is generated for the guid, and the new value for

COOKIES 522 is directory cookies\x 332; the new value for HISTORY 524 is directory history\x 334; and the new value for USER-DEFINED 526 is directory user-defined\x 338.

If the identity has been used before, then the registry subkey values for COOKIES 522, HISTORY 524 and USER-DEFINED 526 are set to values previously stored for the identity. If, for example, a request is submitted to switch to identity 2 316 and identity 2 316 has been used before, subkey COOKIES 522 is set to reference directory cookies\2 342; subkey HISTORY 524 is set to reference directory history\2 346; and subkey USER-DEFINED 526 is set to reference directory user-defined\2 350.

Fig. 6 is a flow diagram outlining management of shared and user-specific Internet content. At step 600, a user enters a universal resource locator (URL) as the user is browsing the Internet. A determination is made at step 602 as to whether Internet content that the user desires to download is stored in the cache. If the content is not in the cache ("NO" branch, step 602), then the content is downloaded from the Internet at step 604.

The URL of the content is hashed at step 606 to determine an index entry that indicates the memory location in which the content will be stored. In memory, a URL is stored as a series of bytes. A hash transformation is applied to the value of the first letter in the URL and the value is stored. The value of the next letter is hashed and the result is combined with the stored hash value. This process repeats until the entire URL has been hashed to result in a URL hash value. This hash value is used to lookup an index location (record) in the index 400 in directory TIF 336.

1 specific content. The pointers are then set to this memory location and the
2 browsing continues at step 618.

3 4 Conclusion

5 Utilizing the described implementations for managing Internet context,
6 switching Internet contexts can be accomplished without a current user having to
7 first log off and a second user log on. To a user, it simply appears as if the browser
8 has been restarted to begin a new session without having terminated the browser.
9 Internet data that is specific to particular users is managed so that the browser can
10 better identify an Internet context with a particular identity, or user.

1 **CLAIMS**

2

3 1. A method, comprising:

4 receiving a request to switch from a current network context to a new

5 network context; and

6 switching from the current network context to the new network context

7 without process shutdown.

8

9 2. The method as recited in claim 1, wherein process shutdown includes

10 terminating a user session utilizing the current network context and logging into a

11 user session utilizing the new network context.

12

13 3. The method as recited in claim 1, wherein the current network

14 context includes web page data specific to a user, and wherein the web page data

15 is stored in a memory location based on a hash of a universal resource locator

16 (URL) for the web page.

1
2 **8.** The method as recited in claim 4, wherein the switching to a new
3 network content further comprises:

4 storing the current network context in a directory uniquely associated with
5 the current guid.
6

7 **9.** The method as recited in claim 1, wherein the current network
8 context is a current Internet context and the new network context is a new Internet
9 context.
10

11 **10.** The method as recited in claim 1, further comprising:
12 determining if the new network context is valid; and
13 switching network contexts only if the new network context is valid.
14

15 **11.** The method as recited in claim 1, wherein switching network
16 contexts further comprises switching universal resource locator (URL) cache
17 components from current URL cache components to new URL cache components.
18

19 **12.** The method as recited in claim 1, wherein the new network context
20 is a default network context.
21

22 **13.** The method as recited in claim 1, wherein a network context
23 comprises a set of objects, one object for each network state.
24
25

1 **14.** The method as recited in claim 1, wherein a network context is an
2 Internet context that comprises a set of objects, one object for each Internet state.
3

4 **15.** The method as recited in claim 14, wherein the set of objects is
5 comprised of one or more of the following types of objects: cookies, history,
6 Internet content, or user-defined data.
7

8 **16.** The method as recited in claim 1, wherein:

9 the network context comprises cache components;

10 the switching further comprises:

11 shutting down cache components of the current network context to
12 prevent operations utilizing the cache components;

13 flushing data uniquely associated with the current network context;

14 creating a set of cache components for the new network context; and

15 resetting a session start time to begin a new session.
16
17
18
19
20
21
22
23
24
25

17. The method as recited in claim 1, wherein:

the current network connection is an Internet connection;

the new network connection is an Internet connection;

the current network context is an Internet context that includes current web page content;

the new network context is an Internet context that includes new web page content;

the method further comprises:

storing the current web page content;

setting one or more global pointers to reference the new web page content; and

the switching further comprises:

utilizing the referenced new web page content for further processing.

18. The method as recited in claim 17, wherein setting one or more global pointers to reference the new web page content further comprises:

hashing a universal resource locator (URL) of a web page from which the web page content is derived; and

setting one or more global pointers to the new web page content in a memory location associated with the hash value derived from hashing the URL.

1
2 **19.** The method as recited in claim 18, wherein the setting one or more
3 global pointers further comprises:

4 identifying the new web page content as being user-specific;
5 determining a globally unique identifier (guid) associated with the new
6 Internet context;
7 determining a value associated with the guid;
8 hashing a combination of the URL and the value associated with the guid;
9 setting the one or more global pointers to the new web page content in a
10 memory location associated with the hash value derived from hashing the
11 combination of the URL and the value associated with the guid.
12

13 **20.** The method as recited in claim 19, wherein the value associated
14 with the guid is an ordinal.
15

16 **21.** A computer-readable medium having computer-executable
17 instructions that, when executed by a computer, perform the following steps:

18 receiving a request to switch from a first Internet context associated with a
19 first identity to a second Internet context associated with a second identity;
20 halting operations utilizing the first Internet context; and
21 initializing operations utilizing the second Internet context without
22 requiring a process shutdown.
23
24
25

1 **22.** The computer-readable medium as recited in claim 21, wherein the
2 halting operations utilizing the first Internet context includes storing first Internet
3 context data in one or more containers associated with the first identity.
4

5 **23.** The computer-readable medium as recited in claim 21, wherein the
6 initializing operations utilizing the second identity includes setting one or more
7 global pointers to Internet context data associated with the second identity that is
8 stored in one or more containers associated with the second identity.
9

10 **24.** The computer-readable medium as recited in claim 21, wherein the
11 initializing operations utilizing the second identity includes setting one or more
12 global pointers to reference default Internet context data and associating the
13 Internet context data with the second identity.
14
15
16
17
18
19
20
21
22
23
24
25

25. An Internet management object stored on a computer-readable medium, comprising computer-executable instructions that, when executed on a computer, perform the following steps:

receiving a request to switch from a first Internet context associated with a first identity to a second Internet context associated with a second Internet context;

storing the first Internet context in one or more containers associated with the first identity;

setting one or more global pointers to reference the second Internet context located in one or more containers associated with the second identity without requiring open processes associated with the first identity to shut down.

26. The Internet management object as recited in claim 25, further comprising computer-executable instructions to perform the following steps:

determining if the second identity has been utilized previously; and

if the second identity has not been utilized previously, creating a new Internet context and setting one or more global pointers to reference the new Internet context stored in `new` containers and associating the new Internet context with the second identity.

27. The Internet management object as recited in claim 25, wherein the first Internet context includes first Internet content stored in a memory location and identified in an index record, the index record being identified according to a hash value of a URL associated with the first Internet content.

1 **28.** The Internet management object as recited in claim 27, wherein the
2 first Internet content is shared content.

3
4 **29.** The Internet management object as recited in claim 25, wherein the
5 first Internet context includes first Internet content stored in a memory location
6 and identified in an index record, the index record being identified according to a
7 hash value of a URL associated with the first Internet content and a value uniquely
8 associated with the first identity.

9
10 **30.** The Internet management object as recited in claim 29, wherein the
11 first Internet content is user-specific content.

12
13 **31.** A computer system, comprising:
14 a registry that includes one or more global pointers that reference one or
15 more containers that store a first Internet context and a second Internet context;
16 an Internet management component that associates a first identifier with the
17 first Internet context and a second identifier with the second Internet context;
18 wherein the Internet management component is configured to halt
19 processing of the first Internet context and initialize processing by the second
20 Internet context without shutting down other processes when it receives a request
21 to switch from the first identity to the second identity.

1 **32.** The computer system as recited in claim 31, wherein:
2 the first Internet context includes first Internet content from a first web page
3 having a first universal resource locator (URL);
4 one of the global pointers references a first memory location derived by
5 hashing the first URL; and
6 the Internet management component is further configured to store the first
7 Internet context data in a container referenced by the global pointer that references
8 the first memory location.

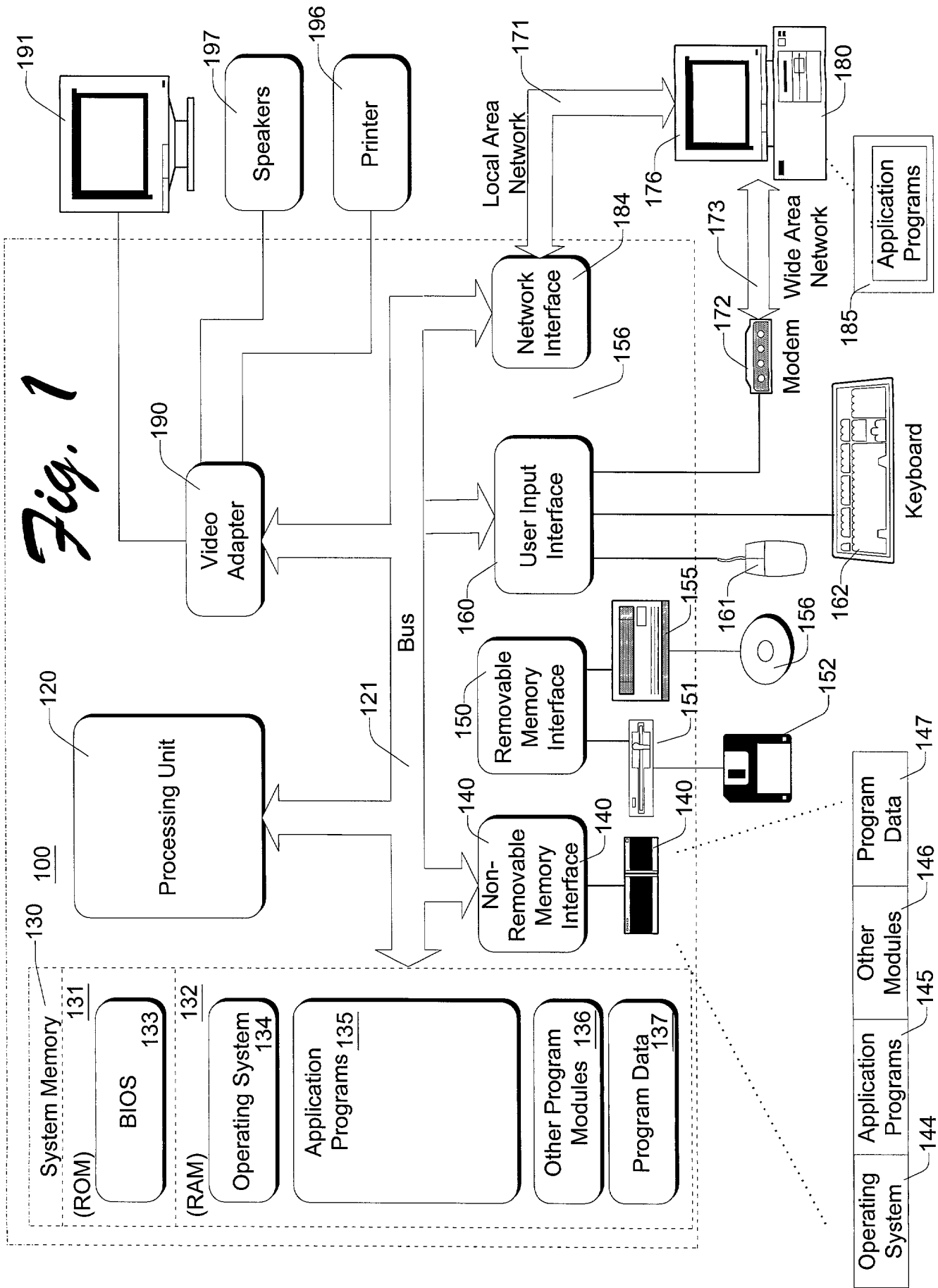
9
10 **33.** The computer system as recited in claim 31, wherein:
11 the first Internet context includes first Internet content from a first web page
12 having a first universal resource locator (URL);
13 the first identity is associated with a unique value;
14 one of the global pointers references a first identity memory location
15 derived by hashing a combination of the first URL and the unique value; and
16 the Internet management component is further configured to store the first
17 Internet context data in a container referenced by the global pointer that references
18 the first identity memory location.

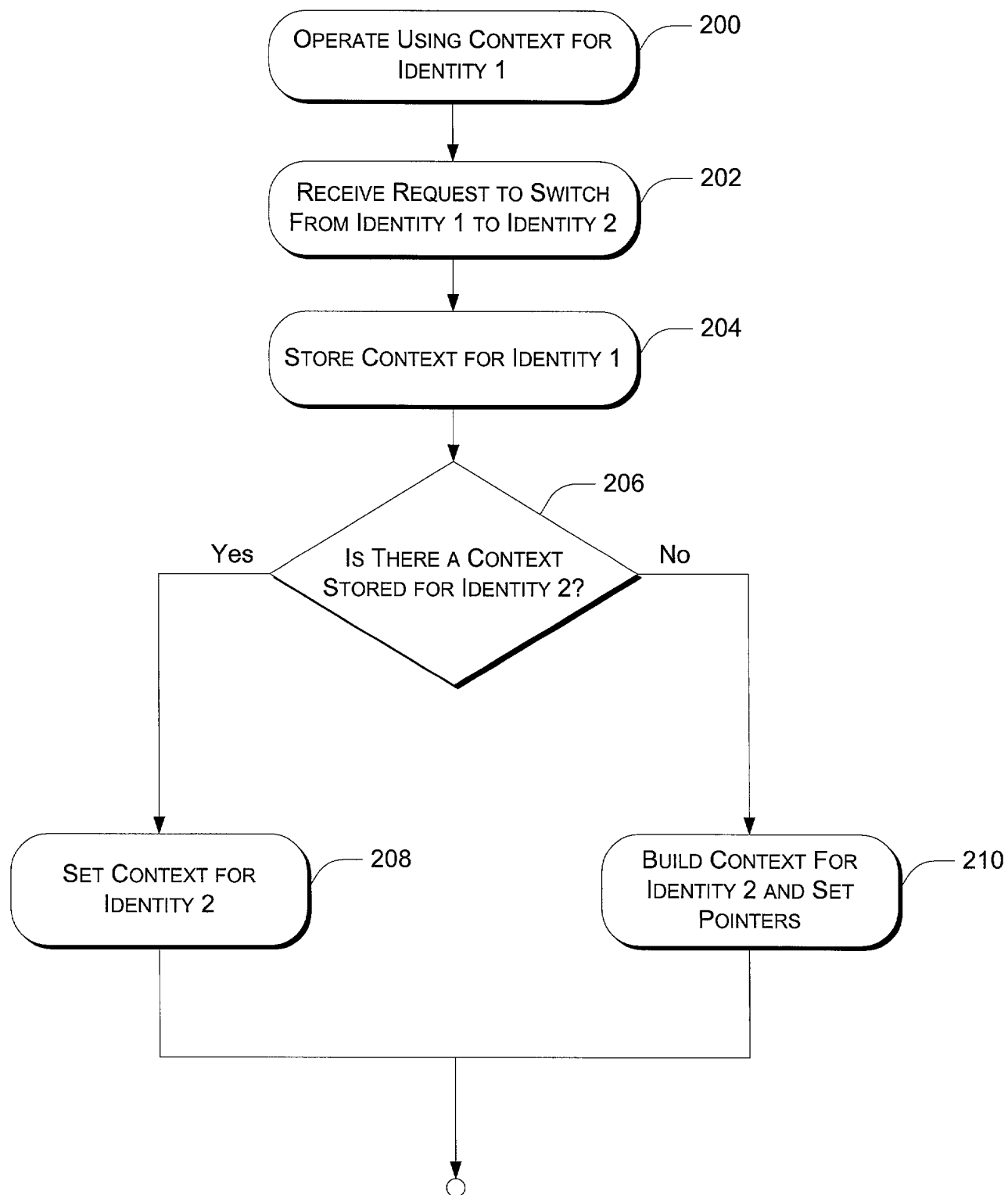
1 **34.** The computer system as recited in claim 31, wherein:
2 the second Internet context includes second Internet content from a second
3 web page having a second universal resource locator (URL);
4 one of the global pointers references a second memory location derived by
5 hashing the second URL; and
6 the Internet management component is further configured to set the global
7 pointer to reference the second memory location.

8
9 **35.** The computer system as recited in claim 31, wherein:
10 the second Internet context includes second Internet content from a second
11 web page having a second universal resource locator (URL);
12 the second identity is associated with a second unique value;
13 one of the global pointers references a second identity memory location
14 derived by hashing a combination of the second URL and the second unique
15 value; and
16 the Internet management component is further configured to set the global
17 pointer to reference the second identity memory location.

1 **ABSTRACT**

2 Systems and methods for switching from a first Internet context to a second
3 Internet context without process shutdown are described. Internet context data,
4 such as cookies, history and user-defined data, is stored in containers unique to
5 each user on a system. Internet content is stored in a common location so
6 redundant downloaded information is not stored. Content information is found or
7 stored by hashing a URL and indexing the memory location according to the
8 resulting hash value. If content data is specific to a particular user, a hash is
9 performed on a combination of the URL and an ordinal associated with the user's
10 unique identity to obtain a hash value unique to the user. The user-specific
11 content is then stored and the memory location is indexed according to the unique
12 hash value.



*Fig. 2*

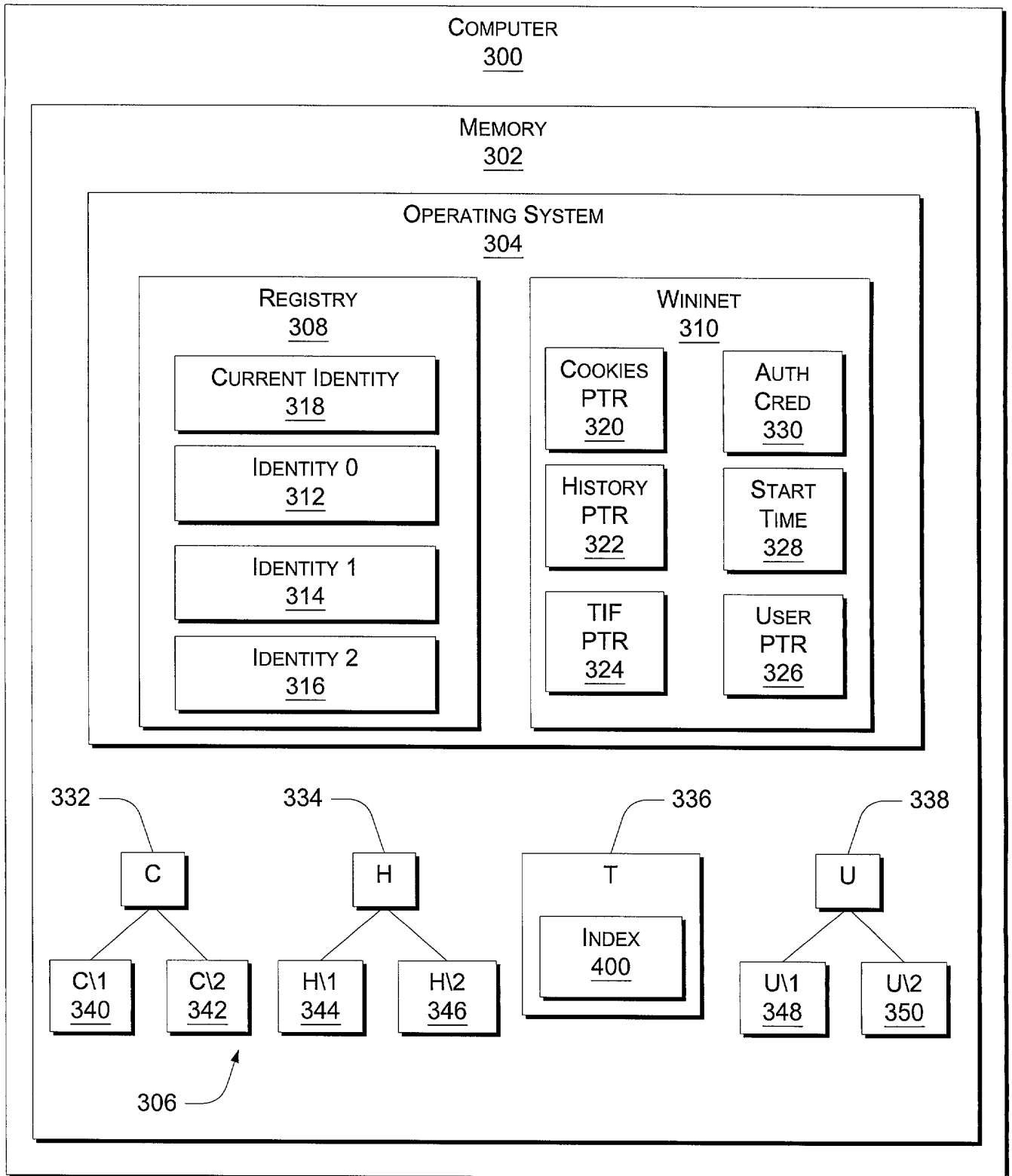


Fig. 3

402a	404	406	408
402b	URL	FILENAME	DATE
402c	URL	FILENAME	DATE
402d	URL	FILENAME	DATE
402n	URL	FILENAME	DATE
	URL	FILENAME	DATE

400

Fig. 4

516 // IDENTITY 2
518 //

Fig. 5

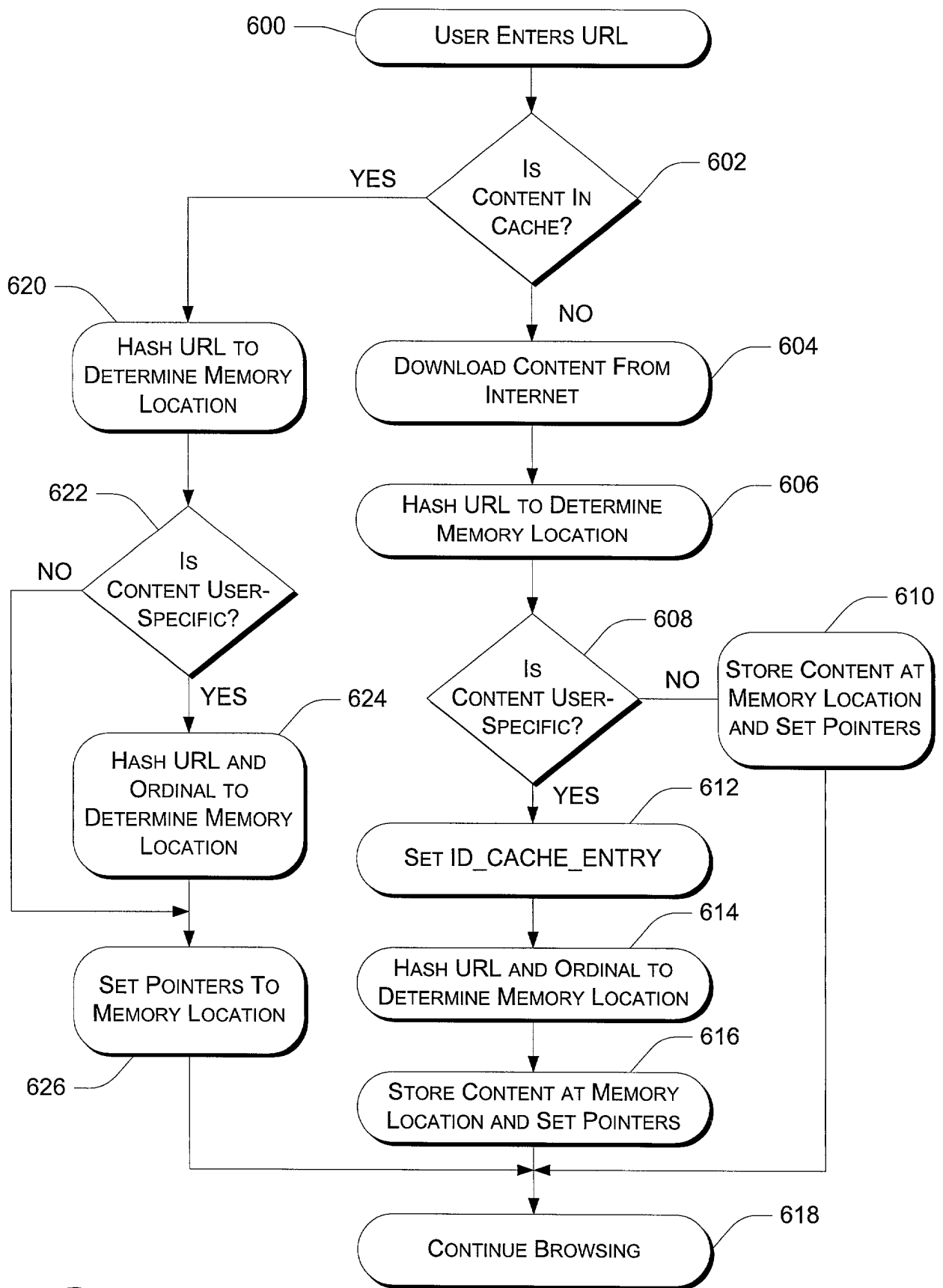


Fig. 6